

HIERARCHICAL EVENT FILTERING FOR STORAGE SERVER

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The invention relates generally to the field of computer systems and, more specifically, to a technique for reducing the number of events that are transmitted by a server host, e.g., in a client/server application programming interface (API).

Description of the Related Art

[0002] Computer systems include host computers that communicate with one another via a network to run network applications. Typically, software is distributed and run on two or more hosts to realize the applications. The network applications have application-layer protocols that define the format and order of the messages that are exchanged between the hosts, and what actions to take when a message is transmitted or received. In particular, a network application typically includes a client side and a server side. In this case, the application may be referred to as a client/server application. A client side on one host may communicate with a server side on another host. The client is usually the host that initiates a communication or session with another host.

[0003] Moreover, in a client/server application programming interface (API), the server generates events and transmits them over a network to the client. The API includes a portion run on the server and a corresponding portion run on the client. An event generally refers to an occurrence of significance to a task, such as the opening of a window or the completion of an asynchronous operation. An event may involve any significant change in the state of a system resource, network resource, or network application. An event can be generated, e.g., for a problem, for the resolution of a

problem, or for the successful completion of a task. Examples of events include the normal starting and stopping of a process, the abnormal termination of a process, and the malfunctioning of a server. For instance, a storage system such as a storage server may run software that generates and communicates such events to clients that store data on the server. The IBM® Enterprise Storage Server (ESS) is an example of such a storage system.

[0004] However, performance can be reduced when many events are transmitted due to the use of processing, storage and transmission resources.

BRIEF SUMMARY OF THE INVENTION

[0005] To overcome these and other deficiencies in the prior art, the present invention provides a technique for reducing the number of events that are transmitted by a server in a client-server API. To achieve this, an end application running at the client is allowed to register for only those events that it requires, and not all events that are generated by the server.

[0006] In a particular aspect of the invention, a method is provided for use at a server host in controlling the communication of different event messages to at least one client host. The method includes associating each of the different event messages with at least one event category in a hierarchy of event categories, associating a particular one of the event categories with the at least one client host, and communicating a particular one of the different event messages to the at least one client host when the particular one of the different event messages is a descendant of the particular one of the event categories.

[0007] In another aspect of the invention, a method is provided for use at a client host in controlling the communication of different event messages thereto from a server host. The method includes communicating a registration request to the server host, wherein, at the server host, each of the different event messages is associated with at least one event category in a hierarchy of event categories, the registration request indicating a particular

one of the different event categories to be associated with the client host, and responsive to the communicating, receiving event messages from the server host that are descendants of the particular one of the different event categories.

[0008] A related apparatus and program storage device are also provided.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] These and other features, benefits and advantages of the present invention will become apparent by reference to the following text and figures, with like reference numbers referring to like structures across the views, wherein:

[0010] Fig. 1 illustrates a client-side host and a server-side host in a computer system;

[0011] Fig. 2 illustrates a hierarchy of events and event categories;

[0012] Fig. 3 illustrates a method for providing event registration; and

[0013] Fig. 4 illustrates a method for performing event distribution.

DETAILED DESCRIPTION OF THE INVENTION

[0014] Fig. 1 illustrates a client-side host 100 and a server-side host 150 in a computer system. The client host 100 includes a processor 110, memory 105 and network interface 115, while the server host 150 similarly includes a processor 160, memory 155 and network interface 165. The network interfaces 115 and 165 communicate with one another via a network 130 such as the Internet. The client-side host 100 may implement a client-side API that comprises a client-side of a command-based API, while the server-side host 150 may implement a server-side API that comprises a server-side of the command-based API. In a command-based API, one or more hosts send commands to one or more other hosts. For example, a command-based API may be used by a storage subsystem host and remote hosts that send data to the storage subsystem host to be backed up. The remote hosts may issue commands to the

storage subsystem host to instruct it to backup their data, or to retrieve previously backed up data, for instance.

[0015] The general operation and configuration of the processors 110, 160, memories 105, 155 and network interfaces 115, 165 is well known in the art and is therefore not described in detail. The components illustrated are provided to assist in understanding the invention. The hosts 100, 150 may be general-purpose computers, workstations, servers, portable devices such as PDAs, or other computer devices. The functionality described herein can be achieved by configuring the hosts 100 and 150 with appropriate software. In one approach, the software comprises an object-oriented software such as Java code that is stored in the memories 105, 155. Any type of programming language may be used. In one approach, the memories 105, 155 are program storage devices that store executable software instructions for execution by the processors 110, 160 in a known manner.

[0016] As indicated, the present invention provides a technique for reducing the number of events that are transmitted from the server host 150 to one or more client hosts 100 in a client-server API. This invention solves the problem by maintaining an event category hierarchy and allowing the end application to register at any point in the hierarchy. When registering for events of a particular event category in the hierarchy, the effect is the same as registering for all events at that level down. In other words, the clients register for all the leaf or descendant events of a specified event category. In one approach, only the leaf nodes of the hierarchy, which are the lowest level nodes, match to real events that can be generated. The invention addresses both event registration and event distribution. The various events that may be generated may be arranged in a hierarchy according to various approaches. In one possible approach, an object-oriented hierarchy is used. This approach can be illustrated conceptually by the following example.

[0017] Fig. 2 illustrates a hierarchy of events and event categories. This example uses fruit types as an illustration. In practice, actual events that are generated, and event categories that are defined, can be arranged accordingly. The tree 200 includes a node for Fruit Events 210 at a top level or hierarchy of the tree. Citrus Events 220 and Berry Events 225 are subcategories of Fruit Events. The nodes for Citrus Events 220 and Berry Events 225 are leaf or descendant nodes of the node for Fruit Events 210. Nodes for an Orange Event 230, Lemon Event 232 and Grapefruit Event 234 are subcategories or descendant nodes of Citrus Events and Fruit Events 210, as well as being descendants of the Fruit Events 210. Nodes for a Strawberry Event 236 and a Blueberry Event 238 are descendant nodes of the node for Berry Events 225, as well as being descendants of the Fruit Events 210. The Fruit Events 210, Citrus Events 220 and Berry Events 225 are event categories, while the Orange Event 230, Lemon Event 232, Grapefruit Event 234, Strawberry Event 236 and Blueberry Event 238 are actual events that can be generated.

[0018] For example, a client that has registered for the Orange Event 230 would only get Orange events and not Lemon Events. In this case, the client receives the event message for which the client registered. A client that has registered for the category of Citrus Events 220 would receive the Orange Event 230, Lemon Event 232, and the Grapefruit Event 234, but would not receive the Strawberry Event 236 or the Blueberry Event 238. A client that has registered for the category of Fruit Events 210 would receive the Orange, Lemon, Grapefruit, Strawberry and Blueberry events. In these cases, the client receives event messages for nodes that are descendants of the particular node or event category that is associated with the client. The hierarchy can have one or more levels of event categories.

[0019] In another possibility, the event hierarchy represents an importance or priority hierarchy. For example:

[0020] All events at a top level;

[0021] Subcategories of All events would be high, medium, and low importance;

[0022] High has subcategories X, Y, Z;

[0023] Medium has subcategories A, B, C; and

[0024] Low has subcategories 1, 2, 3.

[0025] Fig. 3 illustrates a method for providing event registration. At block 300, the client host sends a request to the server host to register for a specified event category. For example, there may be five levels of events identified by the number one through five. If a client host registers at a particular level in the hierarchy that is associated with a particular node, then all events of types listed in the branch below that node that are generated will be communicated to the client host. Thus, events in different branches or higher branches of event categories are not provided to the client host. The end application at the client host registers with the server host for a type of event, or event category, that exists in the hierarchy. The registration request is transmitted from the client host to the server host. At block 310, the server host associates the client host with the specified event category. At block 320, the process of blocks 300 and 310 is repeated for additional client hosts and/or applications. The process ends at block 330.

[0026] Fig. 4 illustrates a method for performing event distribution. At block 400, when the server host generates an event, it compares the hierarchical level or node of the generated event with the hierarchical level or node of the event category that has been associated with the client host (block 410), e.g., the event category for which the client registered. If the node for the generated event is the same as, or is a descendant node of, the event category for which the client host registered (block 420), the server host transmits the event message to the client host or application (block 430). Otherwise, the event message is not transmitted to the client host. At block 440, the process of blocks 410, 420 and 430 may be repeated if there are additional client hosts and/or applications. In summary, with event distribution, the server generates an event of a given type or category. For each client, the server determines if the client is associated with that event

category or any higher-level event category in the hierarchy. If an association is found, the event is transmitted to the client. Otherwise, the event is not transmitted.

[0027] The invention has been described herein with reference to particular exemplary embodiments. Certain alterations and modifications may be apparent to those skilled in the art, without departing from the scope of the invention. The exemplary embodiments are meant to be illustrative, not limiting of the scope of the invention, which is defined by the appended claims.